

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
Nobutaka ISHIDERA)
) Group Art Unit: Unassigned
Serial No.: To be assigned)
) Examiner: Unassigned
Filed: February 6, 2001)



For: **SOFTWARE PROCESSING APPARATUS AND RECORDING MEDIUM
ON WHICH PROGRAM IS RECORDED**

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-301636
Filed: October 2, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: February 6, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W., Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE
JAPANESE GOVERNMENT



This is to certify that the annexed is a true copy of the following
application as filed with the Office.

Date of Application : October 2, 2000

Application Number : Patent Application No. Heisei 2000-301636

Applicant (s) : FUJITSU LIMITED

December 22, 2000

Commissioner, Kohzoh Oikawa
Patent Office

Certificate No. Toku 2000-3108356

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

JCS18 U.S. PTO
09/778088
02/07/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
in this Office.

願 年 月 日
Date of Application:

2000年10月 2日

願 番 号
Application Number:

特願2000-301636

願 人
Applicant(s):

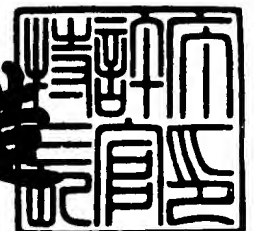
富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年12月22日

特許庁長官
Commissioner,
Patent Office

及川耕造





【書類名】 特許願

【整理番号】 0051446

【提出日】 平成12年10月 2日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/32

【発明の名称】 ソフトウェア処理装置及び記録媒体

【請求項の数】 10

【発明者】

【住所又は居所】 東京都稲城市大字大丸 1 4 0 5 番地 株式会社富士通
パソコンシステムズ内

【氏名】 石寺 紳高

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100079359

【住所又は居所】 東京都港区西新橋 3 丁目 2 5 番 4 7 号 清水ビル 8 階

【弁理士】

【氏名又は名称】 竹内 進

【電話番号】 03(3432)1007

【選任した代理人】

【識別番号】 100093584

【住所又は居所】 東京都港区西新橋 3 丁目 2 5 番 4 7 号 清水ビル 8
階

【弁理士】

【氏名又は名称】 宮内 佐一郎

【電話番号】 03(3432)1007

【手数料の表示】

【予納台帳番号】 009287

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704823

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ソフトウェア処理装置及び記録媒体

【特許請求の範囲】

【請求項 1】

省電力を必要とする動作環境か否かを判断する動作環境判断部と、
省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の環境では低負荷な処理を実行する切替処理部と、
を備えたことを特徴とするソフトウェア処理装置。

【請求項 2】

請求項 1 記載のソフトウェア処理装置に於いて、前記動作環境判断部は、外部から供給される電源によって動作している状態を前記第 1 の環境と判断し、バッテリーによって動作している状態を前記第 2 の環境と判断することを特徴とするソフトウェア処理装置。

【請求項 3】

請求項 1 記載のソフトウェア処理装置に於いて、前記高負荷な処理と前記低負荷な処理は処理装置によって行われ、該処理装置は処理負荷に応じて動作クロック周波数を変動させることを特徴とするソフトウェア処理装置。

【請求項 4】

システムの動作環境を判断する動作環境判断部と、
前記動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り替える切替処理部と、
を備えることを特徴とするソフトウェア処理装置。

【請求項 5】

コンピュータに実行させるためのプログラムであって、
省電力を必要とする動作環境か否かを判断する動作環境判断ステップと、
省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の環境では低負荷な処理を実行する切替処理ステップと、
を備えるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 6】

請求項 5 記載の記録媒体に於いて、前記動作環境判断ステップは、外部から供給される電源によって動作している状態を前記第 1 の環境と判断し、バッテリーによって動作している状態を前記第 2 の環境と判断することを特徴とする記録媒体。

【請求項 7】

請求項 5 記載の記録媒体に於いて、前記低負荷な処理は、前記高負荷な処理が簡略化された処理であることを特徴とする記録媒体。

【請求項 8】

請求項 5 記載の記録媒体に於いて、前記高負荷な処理と前記低負荷な処理は前記コンピュータの処理装置によって行われ、該処理装置は処理負荷に応じて動作クロック周波数を変動させることを特徴とする記録媒体。

【請求項 9】

請求項 5 記載の記録媒体に於いて、前記プログラムは他のプログラムにより共用されることを可能とし、他のプログラムからの通知に応じて前記高負荷な処理と前記低負荷な処理とを切り替えることを特徴とする記録媒体。

【請求項 10】

コンピュータに実行させるためのプログラムであって、
システムの動作環境を判断する動作環境判断ステップと、

前記動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り替える切替処理ステップと、
を備えるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、バッテリーで駆動される携帯可能なノートパソコン等のソフトウェア処理装置及びコンピュータでプログラムを読み取り可能な記録媒体に関し、特にバッテリー駆動時に省電力化したソフトウェアの処理を実行するソフトウェア処理装置及び記録媒体に関する。

【0002】

【従来の技術】

携帯可能なパーソナルコンピュータであるノートパソコンでは、システム動作の電源として充電式のバッテリーを使用するのが一般的である。この充電式バッテリーには電力容量に制限があるため、携帯時に動作可能な時間が限られている。一般的に、携帯機器の動作時間は、より長いことが要求されており、ノートパソコンにおいても携帯時の動作時間を延長させるべく、各種の省電力技術が用いられている。

【0003】

この省電力技術の1つとして用いられているCPUの動作クロック周波数（以下、単に「CPUクロック」という）の自動調整による省電力は、CPUに要求される処理量に応じ、CPUクロックを必要最低限まで低下させることにより、消費電力量を減少させる技術である。

【0004】

【発明が解決しようとする課題】

ところで、CPUクロックの自動調整による省電力を実施中は、CPUに要求

される処理量が少ないほど、CPUクロック低下による高い省電力効果を得ることができる。

【0005】

つまり、CPUクロックの自動調整による省電力を実施中は、静止画のように描画動作が1回で済む処理ほど省電力効果が高く、逆に、アニメーション等のように画像の描画動作を繰り返すタイプの処理程、省電力効果が低いと言える。

【0006】

しかし、ソフトウェアは、環境によらず常に一定の処理を行っているため、1つのソフトウェアを考えた場合、省電力効果の高い低負荷な処理であるか、あるいは省電力効果の低い高負荷な処理であるかのいずれであり、CPUクロックの自動調整による省電力の実施中に、ソフトウェアそのものによって省電力化を図ることは期待されていない。

【0007】

本発明は、ソフトウェアが省電力が必要な環境か否か判断し、それぞれの環境に最適な動作を実現するソフトウェア処理装置及びそのプログラムを格納した記録媒体を提供することを目的とする。

【0008】

【課題を解決するための手段】

図1は本発明の原理説明図である。本発明のソフトウェア処理装置は、ソフトウェアの処理、例えばCPUメータ等のアプリケーション・プログラムを実行する際に、省電力を必要とする動作環境か否かを判断する動作環境判断部48と、省電力を必要としない第1の環境では高負荷な処理を実行し、省電力を必要とする第2の環境では低負荷な処理を実行する切替処理部50とを備えたことを特徴とする。動作環境判断部48は、外部から供給される電源によって動作している状態を第1の環境と判断し、バッテリーによって動作している状態を第2の環境と判断する。

【0009】

このため本発明によれば、外電源によって動作している省電力を必要としない

動作環境では、通常通りソフトウェアの全機能を有効とした処理を行い、バッテリーによって動作している省電力を必要とする動作環境では、本来のソフトウェアの機能を損わない程度に簡略化し、CPUの負荷を下げる。このCPU負荷の低下により、バッテリーによって動作している状態でのCPUクロックの自動調整機能により、CPUクロックが低負荷処理の必要最低限まで低下されることで、消費電力量を減少させる。

【0010】

ここで、低負荷な処理とは、高負荷な処理が簡略化された処理である。また簡略化された処理とは、高負荷な処理での処理内容の一部の処理である。例えば本発明が対象とするソフトウェアは、例えば一定時間のタイマ割込みで動作して複数枚の画像を順次繰り返し描画するアニメーションである。この場合、切替処理部50は、省電力が必要な第1の環境ではアニメーションの一部の画像を有効とした簡略化された低負荷な処理を実行し、省電力が不要な第2の環境ではアニメーションの全ての画像を有効とした高負荷な処理を実行する。

【0011】

アニメーションでは、通常、描画動作の回数が多いほどCPUの負荷が高くなって省電力効果が低下するが、本発明は、省電力が必要な第1の環境では、アニメーションの最後の画像のみを表示する1回の描画動作で済むようにし、CPU負荷を下げてCPUクロックの自動調整による省電力実施中に、ソフトウェアによっても更に省電力化を図ることができる。

【0012】

また簡略化された処理は、高負荷な処理で用いられる被処理データを加工したものである。例えば次のような例がある。

(1) ソフトウェアは浮動小数点演算を実行するプログラムであり、切替処理部50は、省電力が必要な第1の環境では小数点以下を切り捨てて簡略化された比較的low負荷な処理を実行する。

(2) ソフトウェアは動画プログラムであり、切替処理部50は、省電力が必要な第1の環境では画像の間引き又は解像度を落として簡略化された比較的low負荷な処理を実行する。

【 0 0 1 3 】

また簡略化された処理は、高負荷な処理と同機能を実現する他の処理である。例えばソフトウェアはグラフィカルなユーザインタフェースを表示するプログラムであり、切替処理部 5 0 は、省電力が必要な環境ではユーザインタフェースをダイアログ表示に切替えて簡略化された比較的低負荷な処理を実行する。

【 0 0 1 4 】

また、高負荷な処理と低負荷な処理との切り替えを有効または無効に設定する設定部を有する。例えば動作環境判断部 4 8 及び切替処理部 5 0 によるソフトウェア処理の有効と無効を切替え設定するプロパティを設け、ユーザは、必要に応じて通常のソフトウェア動作とするか、省電力化のために簡略化したソフトウェア動作とするかを選択できる。

【 0 0 1 5 】

また高負荷な処理と低負荷な処理は処理装置によって行われ、この処理装置は処理負荷に応じて動作クロック周波数を変動させる。

【 0 0 1 6 】

更に、本発明の別の形態にあっては、システムの動作環境を判断する動作環境判断部 4 8 と、動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り替える切替処理部 5 0 とを備えたソフトウェア処理装置を提供する。

【 0 0 1 7 】

本発明は、更に、ソフトウェア処理のプログラムを記録したコンピュータ読み取り可能な記録媒体を提供もであり、コンピュータに実行させるためのプログラムであっては、

省電力を必要とする動作環境か否かを判断する動作環境判断ステップと、

省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の環境では低負荷な処理を実行する切替処理ステップと、
を備える。このプログラムの詳細は、装置構成の場合と基本的に同じになる。

【 0 0 1 8 】

【発明の実施の形態】

図 2 は、本発明が適用されるハードウェア構成のブロック図であり、携帯可能なパーソナルコンピュータとして知られたノートパソコンを例にとっており、このノートパソコンにあっては、充電式のバッテリーで動作するのが一般的である。なお、本発明の適用対象となる装置は、ノートパソコンに限定されるものでなく、種々の電子機器に適用可能である。

【 0 0 1 9 】

図 2 において、CPU 1 0 からのバス 1 1 に対しては、メモリ 1 2、外部記憶のための HDD（ハードディスクドライブ） 1 4、DSP 1 6、キーボード 1 8、表示コントローラ 2 0 を介したディスプレイ 2 2、モデム 2 4 が設けられている。

【 0 0 2 0 】

PC カードスロット 2 4 にはモデム内蔵の PC カード 2 6 が接続され、PC カード 2 6 をインタフェースとして携帯電話 2 8 を接続することで外部とのデータ通信ができるようにしている。また電源装置 3 0 が設けられており、電源装置 3 0 に対しては AC アダプタ 3 2 と充電式のバッテリー 3 4 が接続可能である。

【 0 0 2 1 】

バッテリー 3 4 は AC アダプタ 3 2 によって電源が供給されている動作状態の時に充電される。また携帯的な使用の際にはバッテリー 3 4 から電源装置 3 0 に対する電源の供給で動作する。電源装置 3 0 は CPU 1 0 からのバス 1 1 に接続されており、AC アダプタ 3 2 からの電源の供給によって動作する動作環境と、バッテリー 3 4 によって動作する動作環境のそれぞれを示す情報を CPU 1 0 側に提供できるようにしている。

【 0 0 2 2 】

CPU 1 0 にバス 1 1 を介して接続したメモリ 1 2 には、ノートパソコンの電源投入時に HDD 1 4 からローディングされて展開されることで、OS 3 8 及び本発明のソフトウェア装置として機能するアプリケーション 3 6 が展開されている。OS 3 8 としては例えばマイクロソフト社のウィンドウズ 9 8 やウィンドウズ 2 0 0 0 などが使用される。

【 0 0 2 3 】

本発明が対象とするアプリケーション 3 6 は例えば CPU メータである。CPU メータは、CPU 1 0 の負荷を監視して数値 (%) 又はグラフ表示するプログラムであり、本発明の CPU メータはプログラムの初期化時に飾り的な動作としてアニメーション表示を行うようにしている。

【 0 0 2 4 】

このアプリケーション 3 6 として提供される CPU メータの初期化時のアニメーション処理について、本発明にあっては装置の動作環境としてバッテリー 3 4 による動作状態の有無をチェックし、バッテリー 3 4 による動作状態を取得した場合には省電力を必要とする動作環境と判断し、CPU メータの初期化時のアニメーション処理に切替えて簡略化された比較的低負荷な処理を実行することで、バッテリーによる動作状態での省電力を図るようにしている。

【 0 0 2 5 】

アプリケーション 3 6 における動作環境を判断するため、OS 3 8 に組み込まれている電源制御デバイスドライバ 4 2 から、電源装置 3 0 に対するバッテリーによる動作状態と、AC アダプタによって下位部電源が供給される動作状態を示す情報を API (アプリケーション・プログラミング・インタフェース) 4 0 を通じて取得することで判断できる。なお、この情報の他に API を利用してバッテリーの状態 (残量、充電中、寿命) も取得できる。

【 0 0 2 6 】

図 3 は、本発明のソフトウェア装置として機能する CPU メータの初期化時に飾り的な動作としてのアニメーション表示を実行する機能構成のブロック図である。

【 0 0 2 7 】

図 3 において、図 2 のメモリ 1 2 に展開されたアプリケーション 3 6 により実行される CPU 上の CPU メータ 4 4 は、画面上に CPU メータ表示を行うための初期化部 4 6 を備えており、また初期化部 4 6 による初期化処理の際に飾り的な動作としてのアニメーション処理を行うアニメーション切替処理部 5 0 を備えている。

【 0 0 2 8 】

またCPUメータ44には、ACアダプタ32による動作状態とバッテリー34による動作状態から動作環境を判断する動作環境判断部48が設けられている。このCPUメータ44における初期化時のアニメーションのため、API40を介してOS38の電源制御デバイスドライバ42から、ACアダプタ32による動作か、バッテリー34による動作を取得できるようにしている。

【0029】

またOS38に設けているファイルドライバ56を介してアニメーション切替処理部50は画像ファイル60に保存しているアニメーション処理に使用する画像を取得し、表示ドライバ58を介してディスプレイ22上にアニメーションを表示できるようにしている。

【0030】

CPUメータ44の初期化部46に組み込まれた動作環境判断部48は、アニメーション切替処理部50の処理を実行する際に、省電力を必要とする動作環境か否かを判断する。具体的には電源制御デバイスドライバ42よりAPI40を介してバッテリー34による動作状態を取得した場合は、省電力を必要とする動作環境（第1の環境）と判断する。

【0031】

また動作環境判断部48は電源制御デバイスドライバ42よりAPI40を介してACアダプタ32による動作状態を取得した場合は、省電力を必要としない動作環境（第2の環境）と判断する。動作環境判断部48の判断結果はアニメーション切替処理部50に与えられる。

【0032】

アニメーション切替処理部50は、動作環境判断部48からバッテリー34による動作状態に基づく省電力が必要な動作環境との判断結果を受けた場合は、低負荷処理部52の処理に切り替え、アニメーションの一部の画像を有効とした簡略化されたCPU10にとって比較的low負荷なlow負荷アニメーション処理を実行する。

【0033】

またアニメーション切替処理部50は、動作環境判断部48からACアダプタ

による動作状態に基づき、省電力が不要な動作環境の判断結果を受けた場合は、高負荷処理部 5 4 に切り替え、アニメーションのために準備している全ての画像を有効とした通常のアニメーション処理、換言すると CPU 1 0 にとって比較的高負荷な高負荷アニメーション処理を実行する。

【 0 0 3 4 】

図 4 は図 3 のアニメーション切替処理部 5 0 で使用する基本画像、高負荷処理となる通常のアニメーション画像、更に省電力のための低負荷処理となるアニメーション画像である。

【 0 0 3 5 】

図 4 (A) はアニメーション処理に使用する基本画像であり、この例では 4 コマの基本画像 6 2, 6 4, 6 6, 6 8 を準備している。これらの基本画像 6 2, 6 4, 6 6, 6 8 は、キャラクターが歩いていく様子を示すアニメーションを作るために使用される。

【 0 0 3 6 】

図 4 (B) は図 4 の 4 コマの基本画像 6 2, 6 4, 6 6, 6 8 を使用した図 3 のアニメーション切替処理部 5 0 で高負荷処理部 5 4 に切り替えた時のアニメーションである。この高負荷処理におけるアニメーションは、図 4 の 4 コマの基本画像 6 2, 6 4, 6 6, 6 8 を順番に 2 回繰り返したアニメーション画像 6 2 - 1, 6 4 - 1, 6 6 - 1, 6 8 - 1, 6 2 - 2, 6 4 - 2, 6 6 - 2, 6 8 - 2 とした 8 コマの画像で構成される。

【 0 0 3 7 】

更に図 4 (C) は図 3 のアニメーション切替処理部 5 0 における低負荷処理における静止画であり、この場合には図 4 (A) の基本画像の中の先頭の基本画像 6 2 のみを静止画像 6 2 - 1 として表示する。もちろん、低負荷処理で使用する静止画としては基本画像の中の任意の 1 コマでよいが、アニメーションの完結を示す意味で最後の画像 6 8 を使用してもよい。

【 0 0 3 8 】

図 5 は、図 3 の CPU メータ 4 4 における初期化部 4 6 のフローチャートである。この CPU メータ 4 4 の初期化処理にあっては、アニメーション切替処理部

50でのアニメーションの実行に必要なパラメータの初期化を行う。まずステップS1で画像番号の初期化を行う。この画像番号は、図7のプログラムリストにあっては「iCounter」で定義されていることから、これを0に初期化する。

【0039】

続いてステップS2で描画フラグを初期化する。この描画フラグが図7のプログラムリストにあっては「fAlready」で定義されており、初期状態にあっては2描画であることから「FALSE」に初期化する。更にステップS3でアニメーションの各画像を描画する周期を決めるタイマの初期化を行う。タイマの初期化としてはn秒間でアニメーション処理を起動するように初期化する。この時間間隔nとしては例えばミリ秒単位の値が使用される。

【0040】

この図5のCPUメータの初期化処理により、ステップS3で設定されたn秒間隔のタイマ割り込みにより図3の動作環境判断部48及びアニメーション切替処理部50によるCPUメータ初期化処理の際の飾り動作としてのアニメーション処理が図6のフローチャートに従って実行される。

【0041】

図6において、CPUメータのアニメーション処理は、まずステップS1でシステム動作環境の取得を行う。具体的には動作環境判断部48がAPI40を介して、電源制御デバイスドライバ42からそのときの電源装置30におけるバッテリー34による動作状態か否か取得する。

【0042】

続いてステップS2でACアダプタ32が接続され、ACアダプタ32から外部電源が供給されている動作状態か否かチェックする。もしACアダプタ32による動作状態でなければバッテリー34による動作状態であることからステップS3に進み、描画フラグ「fAlready」をチェックする。

【0043】

この描画フラグは図5のステップS2で2描画を表す「FALSE」にセットされていることから、描画済みを示す「TRUE」ではなく、従って最初はステ

ップ S 4 に進み、複数準備されているアニメーションのコマ画像の中から図 5 のステップ S 1 で初期化された画像番号 0 の画像を選択して表示し、ステップ S 5 で描画フラグを描画済みである「TRUE」にセットして、バッテリー動作において低負荷となる簡易的なアニメーション処理を 1 回終了する。

【 0 0 4 4 】

続いて再度図 5 のステップ S 3 で初期化されたタイマによる n 秒後のタイマ割り込みで図 6 の処理が起動すると、同様にしてステップ S 1, S 2, S 3 と進んだ後、このとき 1 回目の処理で描画フラグは「TRUE」にセットされていることから、ステップ S 4 の画像番号 0 の表示をスキップし、ステップ S 5 を介して処理を終了する。

【 0 0 4 5 】

この結果、システム動作環境がバッテリーによる動作状態であった場合には、1 回目の CPU メータのアニメーション処理の際に画像番号 0 の静止画を表示するだけであり、その後のタイマ割り込みに対しては画像描画は行わないことになる。

【 0 0 4 6 】

ここでアニメーション処理における CPU 1 0 の負担は、ステップ S 4 における描画処理が 1 回で済むことから、バッテリーによる動作状態にあっては CPU 1 0 の負担が大幅に低減され、省電力化を図ることができる。即ち、バッテリー 3 4 による動作状態にあっては、別のアプリケーションによる処理に基づき CPU クロックの自動調整による省電力処理が実行されており、この CPU クロックの自動調整による省電力処理は CPU 1 0 に要求される処理量に応じ CPU クロックを必要最低限まで低下させることにより消費電力量を減少させている。

【 0 0 4 7 】

このためバッテリー 3 4 による動作状態では、図 6 の CPU メータにおけるアニメーション表示処理で、タイマ割り込みによる最初の 1 回しか画像の描画動作が行われず、この結果、CPU 1 0 に要求される処理量が少なくなり、CPU クロック自動調整機能によってそのクロックを必要最低限まで低下し、消費電力の低減を図ることができる。

【 0 0 4 8 】

一方、ステップ S 2 で A C アダプタ 3 2 による動作状態であった場合には、省電力を必要としない動作環境であることから、ステップ S 6 ～ S 9 により通常のアニメーション処理を実行する。この通常のアニメーション処理はステップ S 6 で、図 5 のステップ S 1 で初期化された画像番号「i C o u n t e r」の画像を表示し、続いてステップ S 7 で画像番号を 1 つインクリメントし、ステップ S 8 で画像番号が予め定めた最大値を超えたか否かチェックする。

【 0 0 4 9 】

最大値を超えていなければステップ S 5 で描画フラグを描画済み「T R U E」にセットし、次の n 秒後のタイマ割り込みを待つ。このようなタイマ割り込みによるステップ S 1, S 2, S 6 ～ S 8, S 5 の繰返しにより、例えば図 4 (B) に示すような順番で画像描画が行われ、アニメーション表示が行われることになる。

【 0 0 5 0 】

このアニメーション処理の実行中にステップ S 8 で画像番号が予め設定した最大値を超えた場合は、ステップ S 9 に進み、画像番号を再び 0 にする再初期化を行う。これによってタイマ割り込みの最大画像番号による繰返しでアニメーション表示が行われることになる。

【 0 0 5 1 】

図 7 は、図 6 の C P U メータにおける初期化時のアニメーション処理を実現するプログラムリストの説明図である。

【 0 0 5 2 】

このプログラムリストにおいて、# a (S P S) はシステム電源状態の格納領域、# b はシステム電源状態の取得部、# c (f A l r e a d y) は内部処理のためのフラグである。また # d はシステム電源状態を判断する動作環境判断部 4 8 の機能を実現する。

【 0 0 5 3 】

また # e 1 ～ # e 4 は簡略化されたアニメーション処理を実行する低負荷処理部 5 2 を実現する。また # f 1 ～ # f 5 は通常時のアニメーション処理であり、

CPU10から見ると高負荷処理部54を構成することになる。更に#gは内部処理のためのフラグ設定処理である。

【0054】

図7のプログラムリストは、CPUへの負荷が非常に高いとされる画像表示処理をバッテリー動作中は、その表示処理を最小限とする動作を実現する。リスト中、#e2及び#f3でコールされる関数「ShowImage」は、画像を画面上に描画するための関数であり、この描画処理が頻繁に呼ばれる程、CPU10への負荷が高い状態にあると言える。

【0055】

図7のプログラムリストで提供されるルーチン「sample」は、図5の初期化処理で初期化したタイマなどにより定期的にコールすることにより、ACアダプタの動作による運用時には、画像番号hBMP[0]～[ANIMATION MAX-1]のパターンをアニメーションさせ、バッテリー接続による運用時はアニメーションを停止し、最後に表示されるアニメーションの1パターンを表示し続けるルーチンである。

【0056】

ここで#aは変数であり、プログラム動作開始時にメモリ領域に確保される。また#cも変数であり、プログラム開始時にメモリ領域が確保され、一度だけ「FALSE（未描画）」に初期化される。更にf2（iCounter）は静的変数であり、プログラム開始時にメモリ領域が確保され、図5の初期化処理により一度だけ初期化され、初期化時の意味は先頭画像番号のイメージを指定する。

【0057】

次に図5のプログラムリストについて処理を説明すると次のようになる。タイマなどによりルーチン「sample」がコールされると、スタック領域に変数#aが確保される。次にシステム動作環境取得処理#bを実行するAPIによりシステム電源の状態が変数#aに格納される。次に判定文#dによりシステムがバッテリーによる動作状態であれば判定文#e1に処理が移行し、システムがACアダプタによる動作状態であれば判定文#f1に処理が移行する。

【0058】

バッテリーによる動作状態に移行する判定文 # e 1 はイメージが 1 つも描画されていないか否かを判定する文であり、判定文 # e 1 の内容が「FALSE（未描画）」であれば # e 2 に処理を移し、「Show Image」によりアニメーション先頭のイメージ h BMP [0] を描画する。

【 0 0 5 9 】

一方、# e 1 の内容が「TRUE（描画済み）」であれば # e 3 に処理を移し、既に描画されていることから # e 4 に移行し、描画処理は何も行わない。各処理を終了すると # g に処理が移る。このような # e 1 ～ # e 4 の処理により、1 回目はアニメーション先頭のイメージを描画するが、それ以降のコールに対してはイメージの更新は行われなくなり、静止画像が表示され続ける。

【 0 0 6 0 】

一方、判定文 # d によりシステムが AC アダプタによる動作で # f 1 に処理が移った場合には、自動的に処理が # f 3 に移る。# f 3 では # f 2 のアニメーションカウンタの値に応じた「n 番目のイメージ」の描画が行われる。

【 0 0 6 1 】

続いて # f 4 に処理が移って # f 2 のアニメーションカウンタが 1 つインクリメントされることにより、次に描画されるイメージを指し示すと同時に # f 2 のアニメーションカウンタと最大値との比較が行われる。このアニメーションカウンタの値が最大値を超えていた場合には # f 5 に処理が移され、# f 2 のアニメーションカウンタに先頭イメージとなる 0 が設定される。

【 0 0 6 2 】

この結果、AC アダプタによる動作時にあっては # f 1 ～ # f 5 により、0, 1, … 最大値, 0, 1, …, 最大値, … の順にイメージが表示されることにより、画像がアニメーションする。

【 0 0 6 3 】

最後の # g にあっては、変数 # c の描画済みフラグに「TRUE」をセットし、画像が表示済みであることが記憶される。

【 0 0 6 4 】

図 8 は、図 7 の # f 1 ～ # f 5 の AC アダプタによる動作時に表示される CP

Uメータ初期化時の飾り動作としてのアニメーションの例である。このCPUメータ初期化時のアニメーションにあつては、図8（A）～（E）の順番にアニメーション表示が繰り返される。

【0065】

図8（A）の画面部分には上段にアプリケーションウィンドウ70が表示され、その下にCPUメータ72が表示されている。CPUメータの初期化時には、アプリケーションウィンドウ70の中に図8（A）～（E）に順番に示すように、コピーライト表記「FUJITSU CPU Meter」となる文字がアプリケーションウィンドウ70の右側からコピーライト表記74-1～74-5のように左側に移動するアニメーション表示を行っている。

【0066】

図9は、図8のCPUメータの初期化時のアニメーションについて、図7のプログラムリストの#e1～#e4でCPU負荷の少ない静止画像を表示した場合である。この図9のバッテリーによる動作状態の時のCPUメータの初期化処理の表示にあつては、CPUメータ表示部72の上のアプリケーションウィンドウ70の中に図8（A）～（E）の中の最後の左隅に移動したコピーライト表記74-5の静止画像が表示される。

【0067】

なお、図4～図7を用いた説明は、バッテリーによる動作状態において、配列要素番号0で示されるイメージ、すなわち、ACアダプタ32から供給される外部電源によって動作しているときに表示されるアニメーションでの先頭イメージを表示するよう設定している。

【0068】

しかし、本例では、この図9の表示を行うために、図6に示すフローチャート、図7に示すプログラムリストにおけるバッテリーによる動作状態の時に表示するイメージとして、最終の配列要素番号で示されるアニメーション表示の最終イメージを表示するよう設定している。

【0069】

図10は、本発明におけるCPUメータ初期化処理の際のアニメーション表示

の他の例である。この場合には、図10 (A) (B) (C) のように、ウィンドウの画面75の左上隅に表示させたCPUメータウィンドウ76-1を時間の経過に伴って図10 (B) (C) におけるCPUメータウィンドウ76-2, 76-3のように画面を横切って右側に移動させていくアニメーションとしている。ここでCPUウィンドウ76-1~76-3の中には、図9と同様なCPUメータ表示部が設けられている。

【0070】

なお、図4~図9においては、アニメーション表示に用いる複数のイメージデータを配列hBMPで定義していたが、本例では、配列hBMPを用いてCPUメータウィンドウを表示するための座標情報を定義する。そして、ShowImage による表示処理の際に、指定される配列hBMPの要素番号で示される座標にCPUメータウィンドウを表示する。これにより、ウィンドウが移動するアニメーションを実現できる。

【0071】

このような図10のCPUメータウィンドウ76-1が画面を横切るアニメーションについて、バッテリーによる動作状態にあっては、図11のように、図10のアニメーションの最後の表示位置を示す図10 (C) の状態でCPUメータウィンドウ76-3が継続して表示されることになる。

【0072】

なお、本例でも図8および図9の例と同様に、バッテリー動作時は配列hBMPの最初の要素番号で示される座標位置ではなく、配列hBMPの最終要素番号で示される座標位置にウィンドウを表示するよう設定している。

【0073】

図12は、CPUメータの初期化処理のアニメーション表示をライブラリ化した場合のライブラリ内部における処理動作のフローチャートである。即ち図5、図6及び図7のCPUメータにおけるアニメーション処理にあっては、アニメーション処理が組み込まれたアプリケーションを例にとるものであったが、図12にあっては、CPUメータの初期化処理をライブラリ化し、その中でアニメーション処理を行うようにしたことを特徴とする。即ち、バッテリー動作時と外部電

源による動作時での処理内容の変更制御を各アプリケーションで共用することができるようにしたものである。

【 0 0 7 4 】

このCPUメータの初期化処理のライブラリ化にあっては、ライブラリ化されているアニメーション表示のためのパラメータをCPUメータの初期化処理のアニメーション表示にバイナリ化するため、次のAPI（アプリケーション・プログラミング・インタフェース）を定義する。

```
void EcoAnimation(HWND hParent, HWND hTarget, HBITMAP hBMP,
                  int iNumImage, UNIT uiInterval)
```

ここで、HWND hParent: 親WindowのWindowハンドル

HWND hTarget: アニメーションを表示する対象WindowのWindowハンドル

HBITMAP hBMP: アニメーションさせる画像の配列へのポインタ

int iNumImage: アニメーションさせる画像の数

UNIT uiInterval: アニメーション間隔（単位 [ms]）

このようにCPUメータの初期化処理をライブラリ化した場合には、上記のように定義したAPIのコールに応じて図12の初期化処理が実行される。

【 0 0 7 5 】

図12の初期化処理にあっては、ステップS1で画像番号の初期化を行い、ステップS2で描画フラグ初期化を行い、更にステップS3でタイマ初期化を行う。このステップS1～S3の初期化は図5のアニメーション処理組込み型の初期化処理と同じであるが、ライブラリ化の初期化処理にあっては、更にステップS5で画像番号最大値の初期化を行い、更に、ステップS5で描画対象ウィンドウを画像パターンを示すhTargetにセットする。そしてライブラリに復帰する。

【 0 0 7 6 】

このようなライブラリ化された初期化処理により、ステップS3によるタイマ初期化で決まる時間毎のタイマ割込みにより、図6のCPUメータのアニメーシ

ョン表示処理が起動し、初期化処理で設定された各パラメータに従ったバッテリーによる動作時の静止画表示、ACアダプタによる動作状態の通常のアニメーション表示が行われる。

【0077】

図13は、本発明におけるCPUメータの初期化処理の際のアニメーション表示につき、バッテリーによる動作状態の時に静止画表示への切替えをユーザが選択して設定するプロパティのフローチャートである。即ち、本発明によるCPUメータの初期化時のアニメーション表示は、バッテリーによる動作状態の時の静止画表示による省電力化の有効無効についてユーザ設定による選択を可能とするため、図13のプロパティの処理フローを準備する。

【0078】

このプロパティによる動作環境によるアニメーション表示の有効無効の切替えは、まずステップS1でダイアログXを起動し、ステップS2でダイアログ選択結果をレジストリY内に保存し、ステップS3で終了化処理を行う。

【0079】

図14は、図13のプロパティの処理動作に伴うプロパティ表示画面である。即ち図13でステップS1のダイアログXを起動すると、処理の動的切替えのプロパティ画面78が表示され、動的切替えを有効とするためにはチェックボックス80をマウスクリックし、「閉じる」ボタン82で設定すればよい。

【0080】

ここで図13のステップS2におけるレジストリYは、OSが持つハードディスク上の設定情報格納領域である。マイクロソフト社のWindows系のOSでは、ソフトウェアに関する設定はレジストリに格納するのが一般的である。レジストリはファイルシステムのような階層構造となっており、特定の情報がレジストリキー（ファイルシステムのパス／ファイル名に相当）により識別される。

【0081】

ここで図13のプロパティによるユーザ設定格納先のレジストリキーとしては例えば次のものが使用できる。

HMY CURRENT USER¥Software¥Fujitsu¥EcoAnimation¥Action

ここで、Actionの値0：環境による動作切替えは行わない。

【0082】

Actionの値1：環境による動作切替えを行う。

図15は図13及び図14のようにプロパティによって処理の動的切替えをユーザが選択的に設定する場合のCPUメータの初期化処理におけるアニメーション処理のフローチャートである。

【0083】

このフローチャートは図6のアニメーション処理のフローチャートにおけるステップS1～S9の先頭にステップS0としてレジストリのチェックルーチンを設けている。ステップS0でレジストリYが1であればシステム環境による動作切替えを行うことから、ステップS1以降の処理に進む。

【0084】

一方、レジストリYが0であった場合にはシステム環境による動作切替えは行わないことから、ステップS6の処理に進むことになる。

【0085】

以上の実施形態はCPUメータの初期化処理における飾り動作としてのアニメーション処理におけるシステム動作環境に応じたCPU低負荷と高負荷の切替えを例にとるものであったが、本発明はこれに限定されず、適宜の装置、ソフトウェアに適用することができる。

【0086】

また以上説明した実施の形態においては、バッテリーによる動作時には省電力を必要とする動作環境とし、外部電源による動作時には省電力を必要しない動作環境と判断するよう構成したが、これに限定されるものではない。

【0087】

例えば、バッテリーの残量が少ない場合を省電力の必要な動作環境、バッテリーの残量が多い場合を省電力の不要な動作環境として定義し、装置がバッテリーで動作している際に、この定義に基づいて動作環境を判断し、その判断結果に基

づいて低負荷処理と高負荷処理とを切り替えるようにしてもよい。なお、このバッテリー残量状況は上述したAPIで取得可能であり、また他の公知の技術でも取得可能なものである。

【0088】

また、通常モードと省電力モードの2つの動作モードを備える装置においては、上述したような動作電源の種類やバッテリー残量の如何に関係なく、通常モードでの動作が指示された場合を省電力が不要な動作環境、省電力モードでの動作が指示された場合を省電力が必要な動作環境と定義し、ユーザによって指定される動作モードに基づいて低負荷処理と高負荷処理とを切り替えるようにしてもよい。

【0089】

なお、この動作モードの取得は、オペレーティングシステムの管理情報から取得するなど公知の技術により可能である。すなわち、本発明は、相対的な2つの状態において、省電力が必要な状態か否かを判断し、それに応じて実行する処理を切り替えるようにすればよいものである。

【0090】

このようなCPUに対する省電力を必要とする動作環境か否かにより高負荷処理と低負荷処理を切替える本発明が対象とするソフトウェアとしては、例えば次のものがある。

【0091】

(1) ソフトウェアは例えば浮動小数点演算を実行するプログラムであり、省電力が必要でない環境では小数点以下を含む浮動小数点演算を実行し、一方、省電力が必要な環境では小数点以下を切り捨てて整数を用いた簡略化された比較的 low 負荷な処理を実行する。

(2) ソフトウェアは動画プログラムであり、切替処理部は省電力が必要でない環境ではテレビのような滑らかな再生画面となる動画の全画像を再生し、一方、省電力の必要な環境では動画のコマを気にならない範囲で間引く間引き処理や、解像度を低下させて簡略化された比較的 low 負荷な処理を実行する。

(3) ソフトウェアは例えばヘルプをMicrosoft社のWordの「イル

カ」のようにグラフィカルなユーザインタフェースを表示するプログラムであり、切替表示部は省電力が必要でない環境ではグラフィカルなユーザインタフェースを表示し、省電力が必要な環境ではユーザインタフェースを簡易型のユーザインタフェース、例えばヘルプを通常のダイアログ表示に切り替えて簡略化された比較的低負荷な処理を実行する。

【 0 0 9 2 】

もちろん、これ以外にも、同じアプリケーションでありながら省電力の必要のない環境での処理に対し、省電力を必要とする環境では一部の機能または簡略化された機能に切り替えてCPUに対する負荷を低減する適宜の処理に本発明は対応することができる。

【 0 0 9 3 】

次に本発明の動作環境に応じて高負荷と低負荷を切り替えるプログラムを格納したコンピュータ読み取り可能な記録媒体の実施形態として、このプログラムは図2の構成をもつコンピュータ上で走るアプリケーションプログラムやライブラリ、その他の制御プログラムとして作成される。

【 0 0 9 4 】

このような本発明の機能を実現するプログラムは、CD-ROM、フロッピーディスク、DVD、光磁気ディスク、ICカードなどの可搬型記憶媒体に格納されて、あるいはモデムやLANインタフェースを利用してデータベースや他のコンピュータシステムからネットワークを介してインストールされる。

【 0 0 9 5 】

インストールされた本発明のプログラムはコンピュータシステムに入力され、そのハードディスクHDDに記憶され、RAMなどを利用してCPUにより実行されることになる。

【 0 0 9 6 】

尚、本発明はその目的と利点を損なわない適宜の変形を含み、また上記の実施形態に示した数値による限定は受けない。

(付記)

(付記 1)

省電力を必要とする動作環境か否かを判断する動作環境判断部と、
省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の環境では低負荷な処理を実行する切替処理部と、
を備えたことを特徴とするソフトウェア処理装置。(1)

(付記 2)

付記 1 記載のソフトウェア処理装置に於いて、前記動作環境判断部は、外部から供給される電源によって動作している状態を前記第 1 の環境と判断し、バッテリーによって動作している状態を前記第 2 の環境と判断することを特徴とするソフトウェア処理装置。(2)

(付記 3)

付記 1 記載のソフトウェア処理装置に於いて、前記低負荷な処理は、前記高負荷な処理が簡略化された処理であることを特徴とするソフトウェア処理装置。

【 0 0 9 7 】

(付記 4)

付記 3 記載のソフトウェア処理装置に於いて、前記簡略化された処理は、前記高負荷な処理での処理内容の一部の処理であることを特徴とするソフトウェア処理装置。

【 0 0 9 8 】

(付記 5)

付記 3 記載のソフトウェア処理装置に於いて、前記簡略化された処理は、前記高負荷な処理で用いられる被処理データを加工したものであることを特徴とするソフトウェア処理装置。

【 0 0 9 9 】

(付記 6)

付記 3 記載のソフトウェア処理装置に於いて、前記簡略化された処理は、前記高負荷な処理と同機能を実現する他の処理であることを特徴とするソフトウェア処理装置。

【 0 1 0 0 】

(付記 7)

付記 1 記載のソフトウェア処理装置に於いて、前記高負荷な処理と前記低負荷な処理との切り替えを有効または無効に設定する設定部を有することを特徴とするソフトウェア処理装置。

【 0 1 0 1 】

(付記 8)

付記 1 記載のソフトウェア処理装置に於いて、前記高負荷な処理と前記低負荷な処理は処理装置によって行われ、該処理装置は処理負荷に応じて動作クロック周波数を変動させることを特徴とするソフトウェア処理装置。(3)

(付記 9)

システムの動作環境を判断する動作環境判断部と、
前記動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り替える切替処理部と、
を備えることを特徴とするソフトウェア処理装置。(4)

(付記 1 0)

省電力を必要とする動作環境か否かを判断する動作環境判断ステップと、
省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の環境では低負荷な処理を実行する切替処理ステップと、
を備えたことを特徴とするソフトウェア処理方法。

【 0 1 0 2 】

(付記 1 1)

付記 1 0 記載のソフトウェア処理方法に於いて、前記動作環境判断ステップは、外部から供給される電源によって動作している状態を前記第 1 の環境と判断し、バッテリーによって動作している状態を前記第 2 の環境と判断することを特徴とするソフトウェア処理方法。

【 0 1 0 3 】

(付記 1 2)

付記 1 0 記載のソフトウェア処理方法に於いて、前記低負荷な処理は、前記高負荷な処理が簡略化された処理であることを特徴とするソフトウェア処理方法。

【 0 1 0 4 】

(付記 1 3)

付記 1 2 記載のソフトウェア処理方法に於いて、前記簡略化された処理は、前記高負荷な処理での処理内容の一部の処理であることを特徴とするソフトウェア処理方法。

【 0 1 0 5 】

(付記 1 4)

付記 1 2 記載のソフトウェア処理方法に於いて、前記簡略化された処理は、前記高負荷な処理で用いられる被処理データを加工したものであることを特徴とするソフトウェア処理方法。

【 0 1 0 6 】

(付記 1 5)

付記 1 2 記載のソフトウェア処理方法に於いて、前記簡略化された処理は、前記高負荷な処理と同機能を実現する他の処理であることを特徴とするソフトウェア処理方法。

【 0 1 0 7 】

(付記 1 6)

付記 1 0 記載のソフトウェア処理方法に於いて、前記切替処理ステップは、有効／無効の設定情報に基づき、前記高負荷な処理と前記低負荷な処理との切り替え処理を行うことを特徴とするソフトウェア処理方法。

【 0 1 0 8 】

(付記 1 7)

付記 1 0 記載のソフトウェア処理方法に於いて、前記高負荷な処理と前記低負荷な処理は処理装置によって行われ、該処理装置は処理負荷に応じて動作クロック周波数を変動させることを特徴とするソフトウェア処理方法。

【 0 1 0 9 】

(付記 1 8)

システムの動作環境を判断する動作環境判断ステップと、
前記動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り

替える切替処理ステップと、
を備えることを特徴とするソフトウェア処理方法。

【 0 1 1 0 】

(付記 1 9)

コンピュータに実行させるためのプログラムであって、
省電力を必要とする動作環境か否かを判断する動作環境判断ステップと、
省電力を必要としない第 1 の環境では高負荷な処理を実行し、省電力を必要とする第 2 の 環境では低負荷な処理を実行する切替処理ステップと、
を備えるプログラムを記録したコンピュータ読み取り可能な記録媒体。(5)

(付記 2 0)

付記 1 9 記載の記録媒体に於いて、前記動作環境判断ステップは、外部から供給される電源によって動作している状態を前記第 1 の環境と判断し、バッテリーによって動作している 状態を前記第 2 の環境と判断することを特徴とする記録媒体。(6)

(付記 2 1)

付記 1 9 記載の記録媒体に於いて、前記低負荷な処理は、前記高負荷な処理が簡略化された処理であることを特徴とする記録媒体。(7)

(付記 2 2)

付記 2 1 記載の記録媒体に於いて、前記簡略化された処理は、前記高負荷な処理での処理内容の一部の処理であることを特徴とする記録媒体。

【 0 1 1 1 】

(付記 2 3)

付記 2 1 記載の記録媒体に於いて、前記簡略化された処理は、前記高負荷な処理で用いられる被処理データを加工したものであることを特徴とする記録媒体。

【 0 1 1 2 】

(付記 2 4)

付記 2 1 記載の記録媒体に於いて、前記簡略化された処理は、前記高負荷な処理と同機能を実現する他の処理であることを特徴とする記録媒体。

【 0 1 1 3 】

(付記 2 5)

付記 1 9 記載の記録媒体に於いて、前記切替処理ステップは、有効／無効の設定情報に基づき、前記高負荷な処理と前記低負荷な処理との切り替え処理を行うことを特徴とする記録媒体。

【0 1 1 4】

(付記 2 6)

付記 1 9 記載の記録媒体に於いて、前記高負荷な処理と前記低負荷な処理は前記コンピュータの処理装置によって行われ、該処理装置は処理負荷に応じて動作クロック周波数を変動させることを特徴とする記録媒体。(8)

(付記 2 7)

付記 1 9 記載の記録媒体に於いて、前記プログラムは他のプログラムにより共用されることを可能とし、他のプログラムからの通知に応じて前記高負荷な処理と前記低負荷な処理とを切り替えることを特徴とする記録媒体。(9)

(付記 2 8)

付記 2 7 記載の記録媒体に於いて、前記プログラムは前記他のプログラムからの通知に含まれる内容に応じて前記高負荷な処理および前記低負荷な処理における処理内容を決定することを特徴とする記録媒体。

【0 1 1 5】

(付記 2 9)

コンピュータに実行させるためのプログラムであって、
システムの動作環境を判断する動作環境判断ステップと、
前記動作環境に応じて、処理装置にとって高負荷な処理と低負荷な処理とを切り替える切替処理ステップと、
を備えるプログラムを記録したコンピュータ読み取り可能な記録媒体。(10)

【0 1 1 6】

【発明の効果】

以上説明してきたように本発明によれば、省電力が必要ない A C アダプタにより外部電源が供給される動作状態の時には、本来の処理機能である C P U 負荷の

高い例えばアニメーションを実行するが、省電力が必要なバッテリーによる動作状態の時には、簡略化された処理である例えば静止画の表示だけが実行され、これによって省電力が必要な運用時においてCPU負荷が低減され、その結果、CPUクロック自動調整によってCPUクロックが低下される期間が増え、このようなソフトウェアにおける動的な省電力の動作環境に適合したCPU負荷の低減により、ノートパソコンなどの充電式のバッテリーを使用する装置の携帯表示の動作時間を更に延長することができる。

【図面の簡単な説明】

【図 1】

本発明の原理説明図

【図 2】

本発明のハードウェア構成のブロック図

【図 3】

本発明の機構構成のブロック図

【図 4】

本発明のアニメーション処理による描画動作の説明図

【図 5】

本発明のアニメーション処理に使用するパラメータの初期化処理のフローチャート

【図 6】

図 2 のアニメーション・ソフトウェアによる本発明の処理動作のフローチャート

【図 7】

図 6 に対応したプログラムリストの説明図

【図 8】

画面上でコピーライト表記がスクロールする本発明の省電力を必要としない環境でのCPU高負荷時となる通常の描画動作の説明図

【図 9】

画面上でコピーライト表示が停止している本発明の省電力を必要とする環境でC

PU低負荷とする描画動作の説明図

【図 1 0】

画面上でアプリケーションのウィンドウが移動する本発明の省電力を必要としない環境でのCPU高負荷時となる通常の描画動作の説明図

【図 1 1】

画面上にアプリケーションのウィンドウが停止している本発明の省電力を必要とする環境でCPU低負荷とする描画動作の説明図

【図 1 2】

本発明のアニメーション・ソフトウェアのパラメータ初期化処理をライブラリ化した場合のフローチャート

【図 1 3】

本発明によるCPU負荷切替えの有効と無効をプロパティにより設定する場合のフローチャート

【図 1 4】

図 1 1 に従ったCPU負荷切替えの有効と無効を設定するプロパティ画面の説明図

【図 1 5】

図 1 2 のプロパティ画面による設定したレジストリの内容に従った本発明によるアニメーション・ソフトウェア処理のフローチャート

【符号の説明】

1 0 : CPU

1 1 : バス

1 2 : メモリ

1 4 : HDD

1 6 : DSP

1 8 : キーボード

2 0 : 表示コントローラ

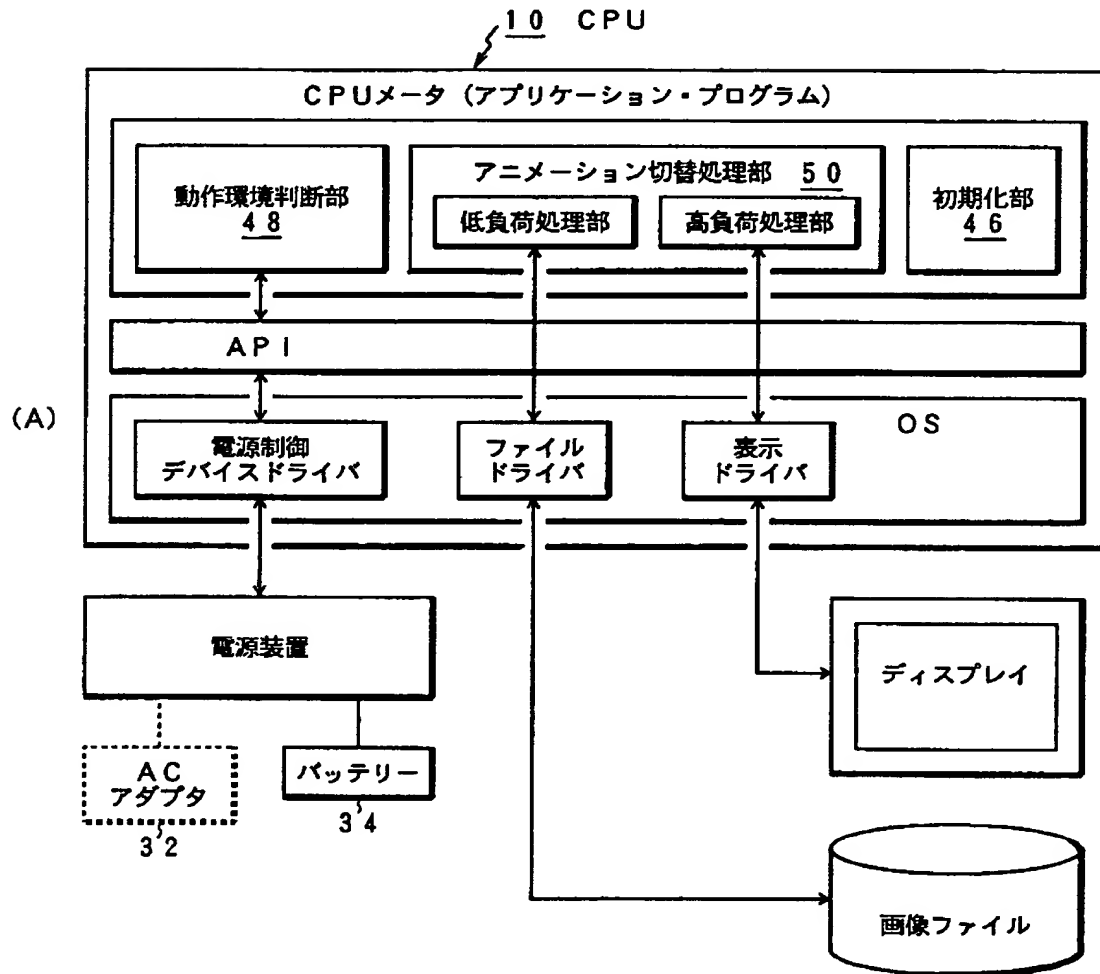
2 2 : ディスプレイ

24 : モデム
26 : PCカード
28 : 携帯電話
30 : 電源装置
32 : ACアダプタ
34 : バッテリー
36 : アプリケーション
38 : OS
40 : API
42 : 電源制御デバイスドライバ
44 : アニメーション・ソフトウェア
46 : 動作環境判断部
48 : アニメーション切替処理部
50 : 低負荷処理部
52 : 高負荷処理部
54 : 初期化部
56 : ファイルドライバ
58 : 表示ドライバ
60 : 画像ファイル
62, 54, 66, 68 : 画像
70 : アプリケーション・ウィンドウ
72 : CPUメータ
74-1~74-5 : コピーライト表示
75 : 初期画面
76-1~76-3 : アプリケーション・ウィンドウ
78 : プロパティ画面
80 : チェックボックス
82 : 閉鎖ボタン

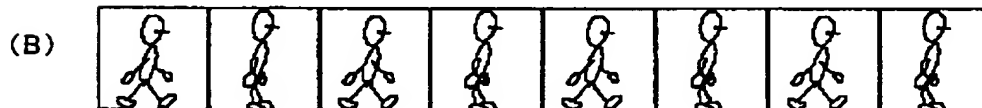
【書類名】 図面

【図 1】

本発明の原理説明図



アニメーション (高負荷)

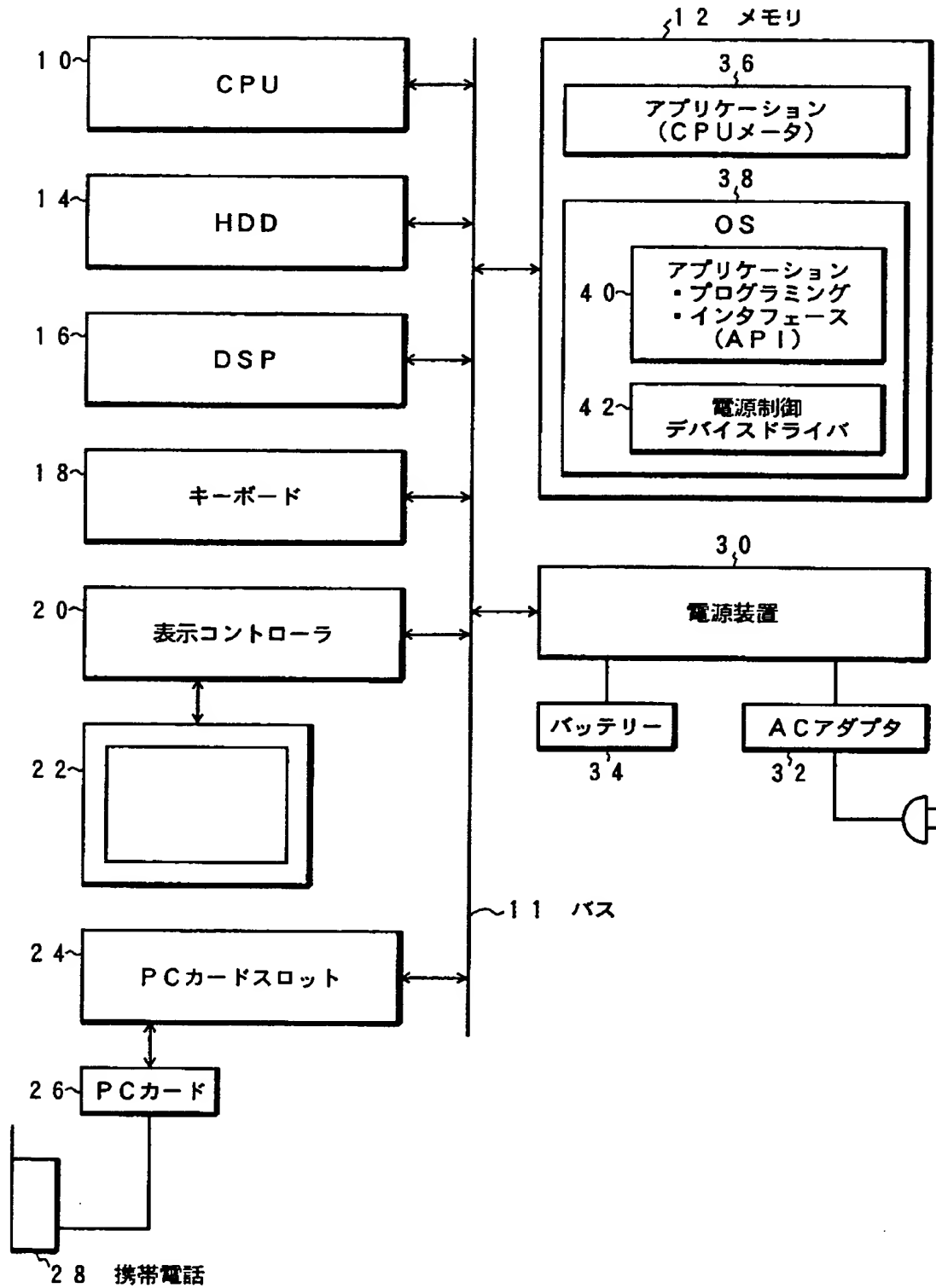


静止画 (低負荷)



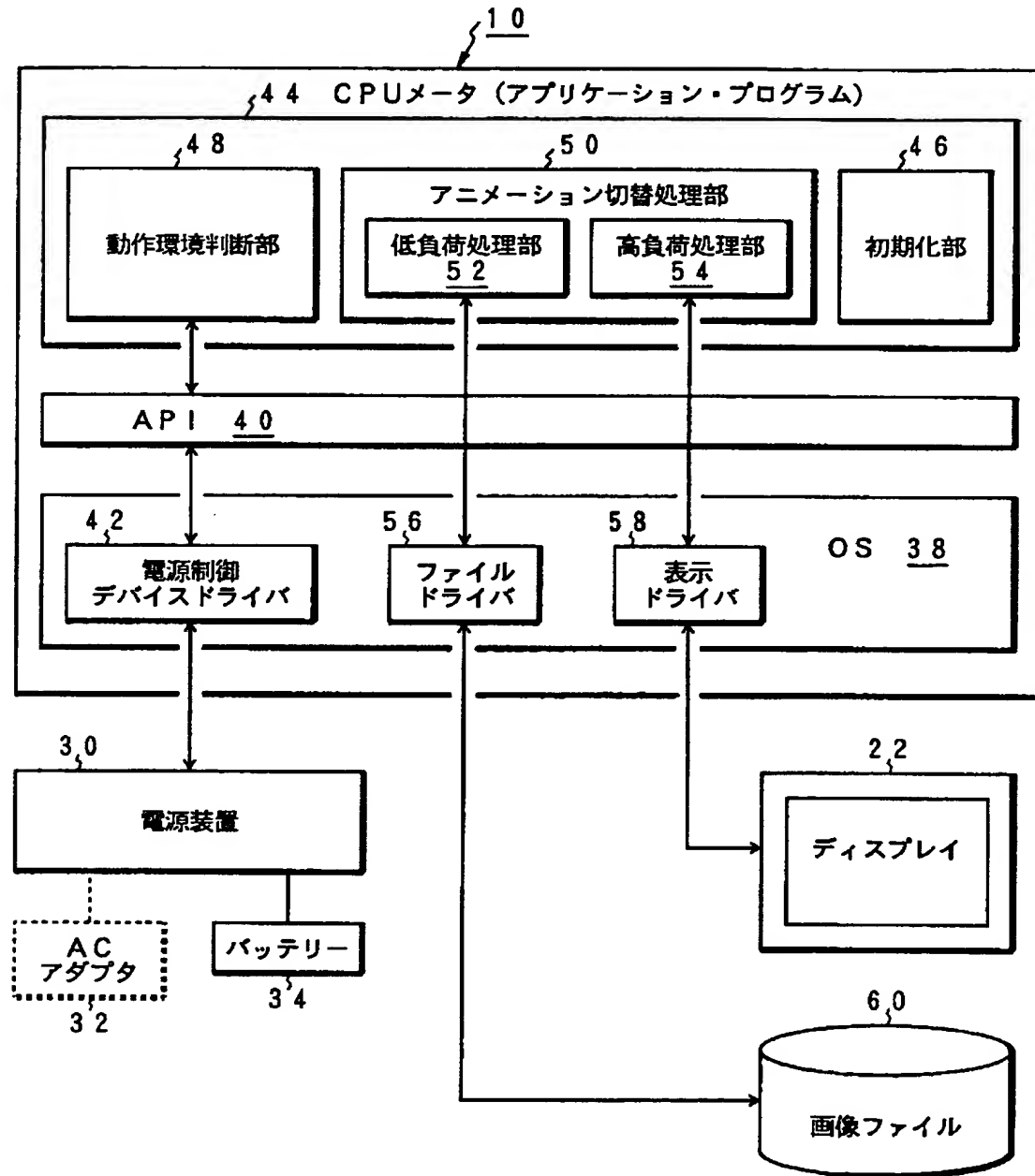
【図 2】

本発明のハードウェア構成のブロック図



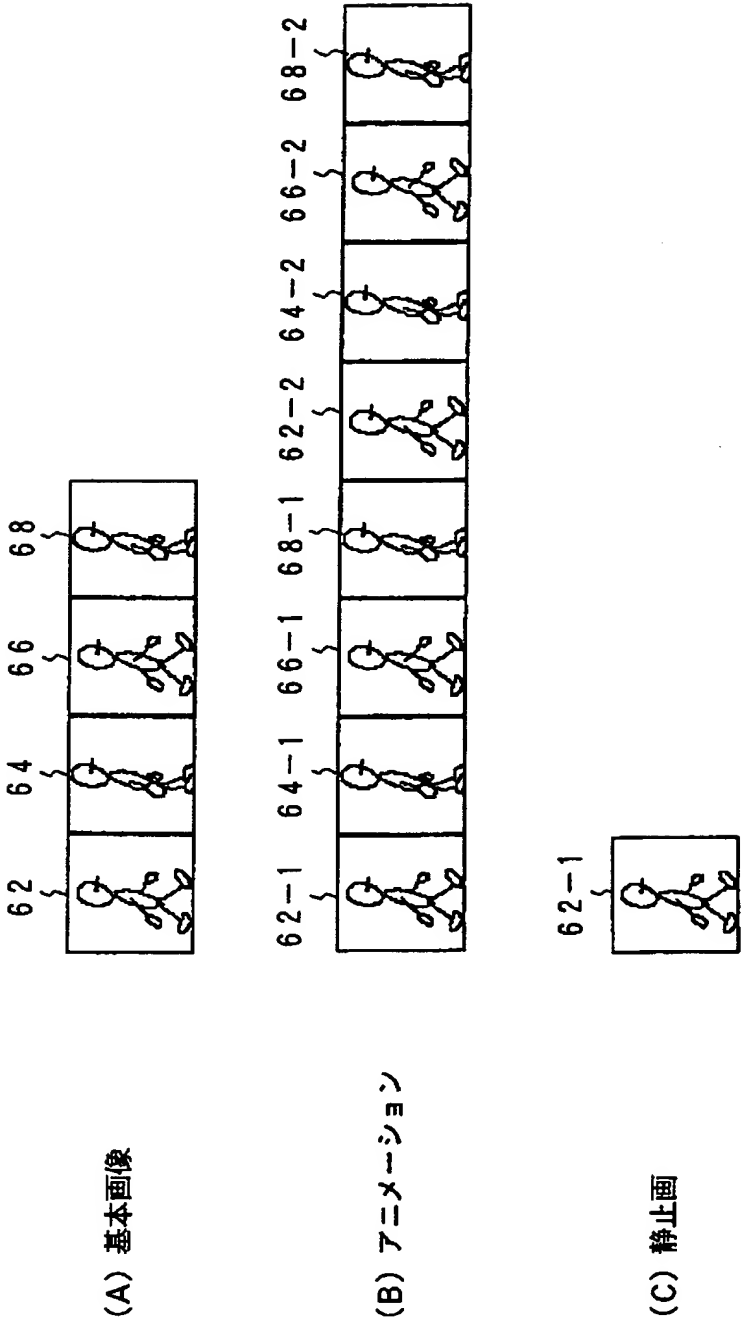
【図 3】

本発明の機構構成のブロック図



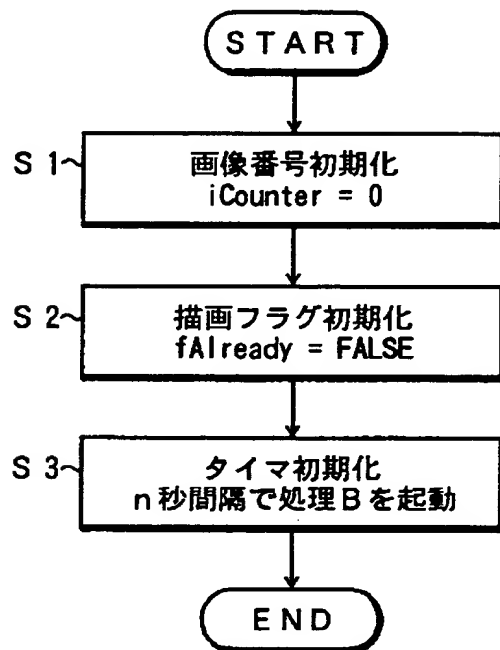
【図 4】

本発明のアニメーション処理による描画動作の説明図



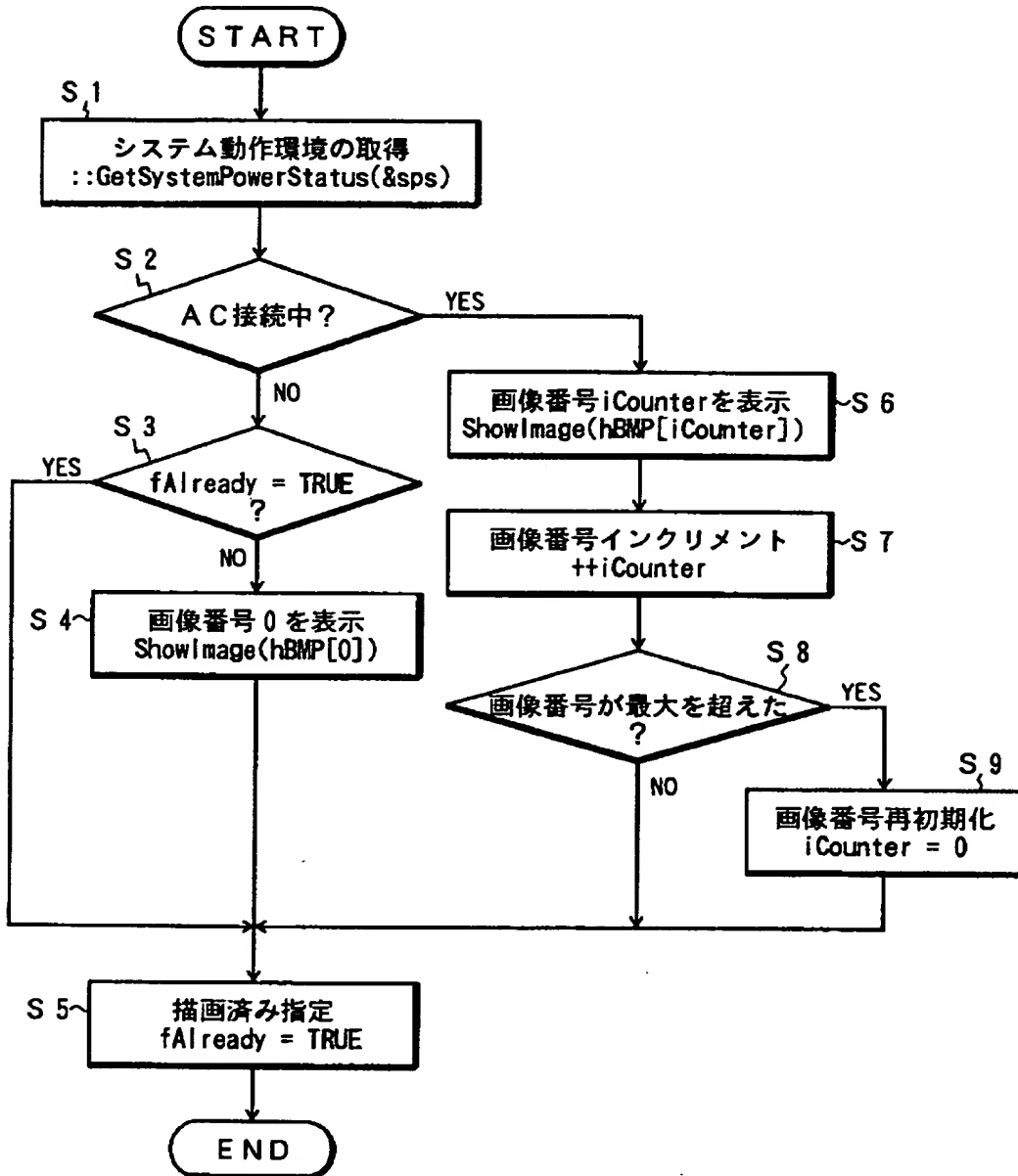
【図 5】

本発明のアニメーション処理に使用するパラメータの
初期化処理のフローチャート



【図 6】

図 2 のアニメーション・ソフトウェアによる本発明の処理動作のフローチャート



【図 7】

図 6 に対応したプログラムリストの説明図

List.1

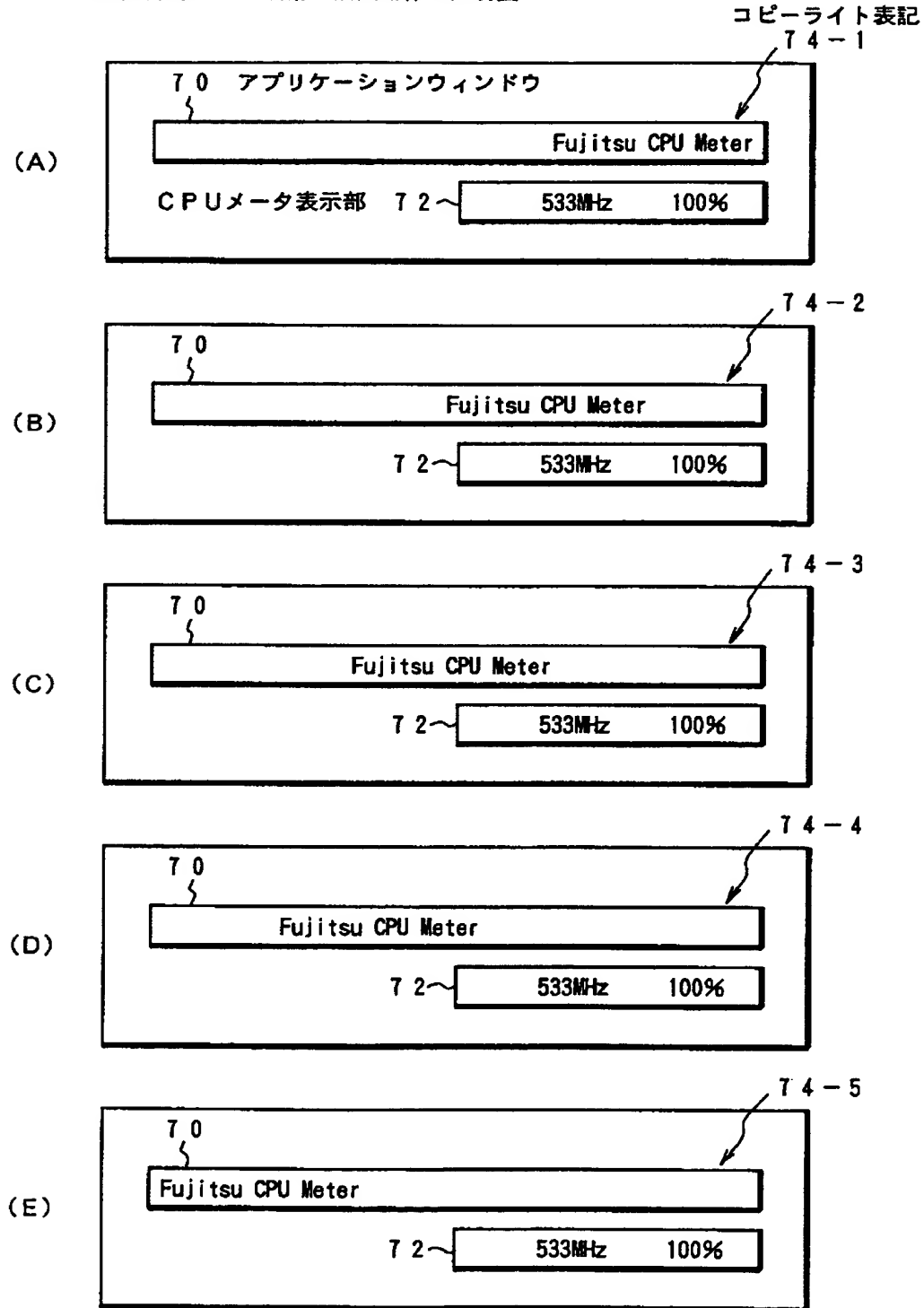
```

void sample()
{
    //-----
    // #a システム状態保存領域
    //-----
    SYSTEM-POWER-STATUS sps;
    //-----
    // #b システム動作環境取得
    //-----
    ::GetSystemPowerStatus( &sps );
    //-----
    // #c 描画済みフラグ (1 度だけ初期化される)
    //-----
    static BOOL fAlready = FALSE;
    //-----
    // #d システムがバッテリー動作中か?
    //-----
    if ( AC-LINE-OFFLINE! = sps.ACLineStatus ) {
        //-----
        // #e1 描画されていない
        //-----
        if ( !fAlready ) {
            //-----
            // #e2 描画する
            //-----
            ShowImage( hBMP[0] );
            //-----
            // #e3 既に描画されている
            //-----
        } else {
            //-----
            // #e4 何もしない
            //-----
        }
        //-----
    }
    //-----
    // #f1 システムが A C アダプタ動作中
    //-----
    //-----
    } else {
        //-----
        // #f2 アニメーションカウンタ (1 度だけ初期化される)
        //-----
        static int iCounter = 0;
        //-----
        // #f3 描画する
        //-----
        ShowImage( hBMP[i] );
        //-----
        // #f4 アニメーションカウンタインクリメント
        //-----
        if ( ANIMATION-MAX < ++iCounter ) {
            //-----
            // #f5 最後まで描画済みであれば、最初に戻る
            //-----
            iCounter = 0;
        }
        //-----
    }
    //-----
    // #g 描画済み
    //-----
    fAlready = TRUE;
    return;
}

```

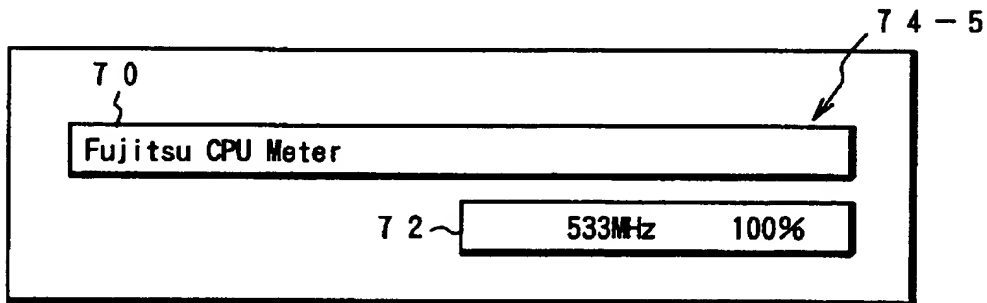
【図 8】

画面上でコピーライト表記がスクロールする本発明の省電力を必要としない環境での
CPU高負荷時となる通常の描画動作の説明図



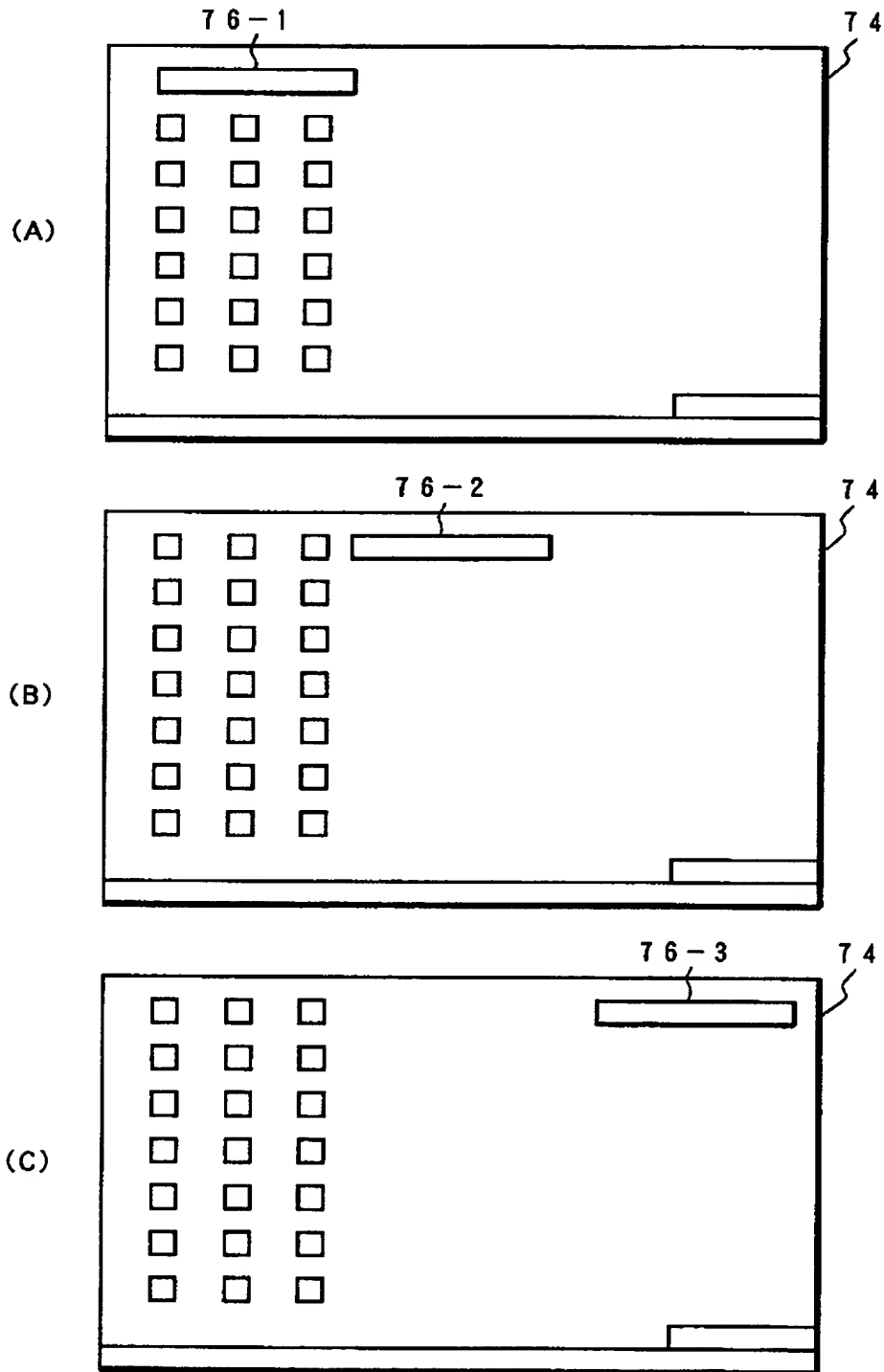
【図 9】

画面上でコピーライト表記が停止している本発明の省電力を必要とする
環境で CPU 低負荷とする描画動作の説明図



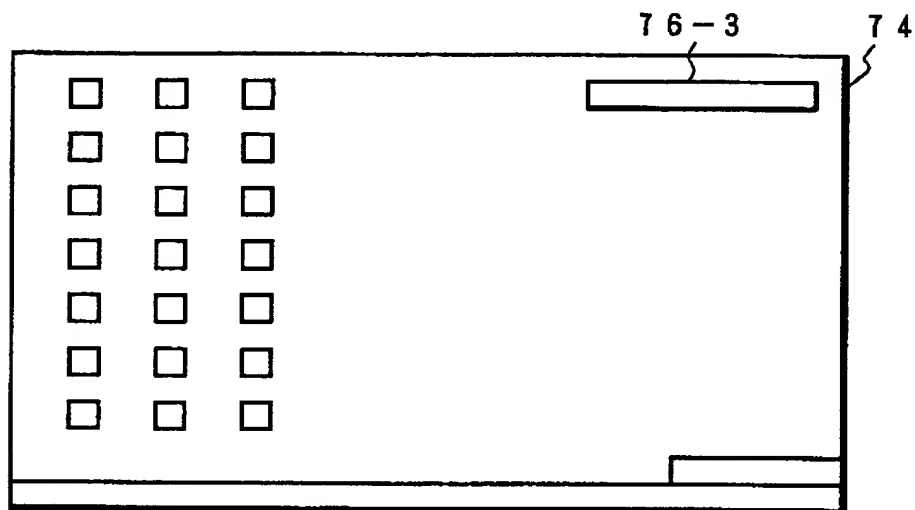
【図 1 0】

画面上でアプリケーションのウィンドウが移動する本発明の省電力を必要としない
環境でのCPU高負荷時となる通常の描画動作の説明図



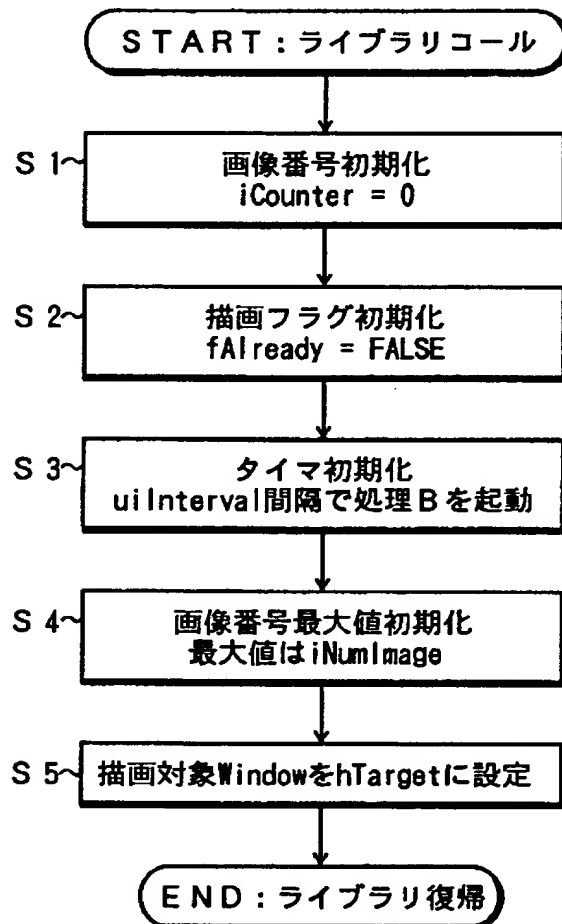
【図 1 1】

画面上にアプリケーションのウィンドウが停止している本発明の省電力を
必要とする環境でCPU低負荷とする描画動作の説明図



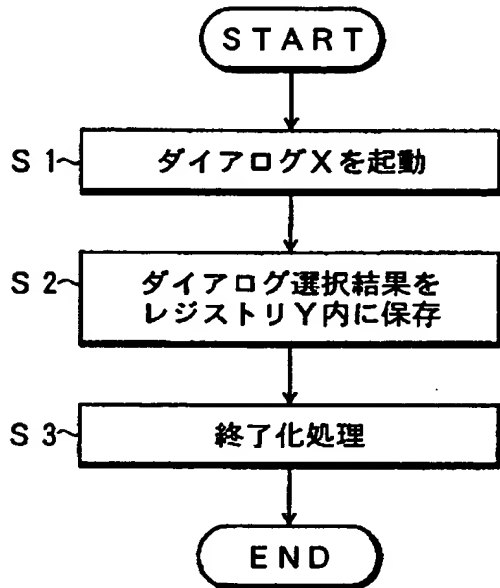
【図 1 2】

本発明のアニメーション・ソフトウェアのパラメータ初期化処理
をライブラリ化した場合のフローチャート



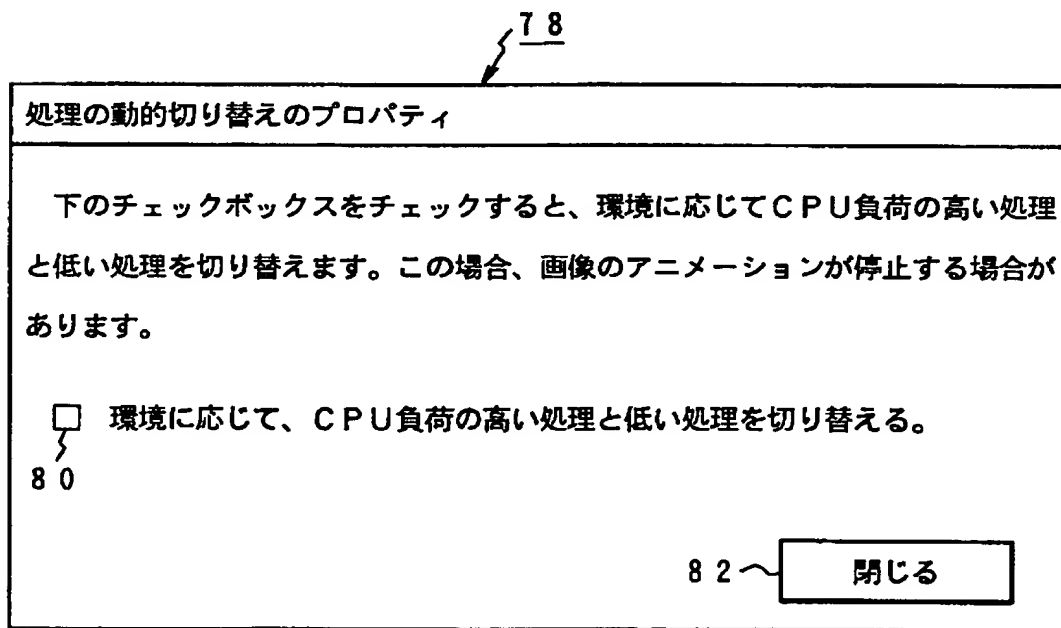
【図 13】

本発明によるCPU負荷切替えの有効と有効をプロパティ
により設定する場合のフローチャート



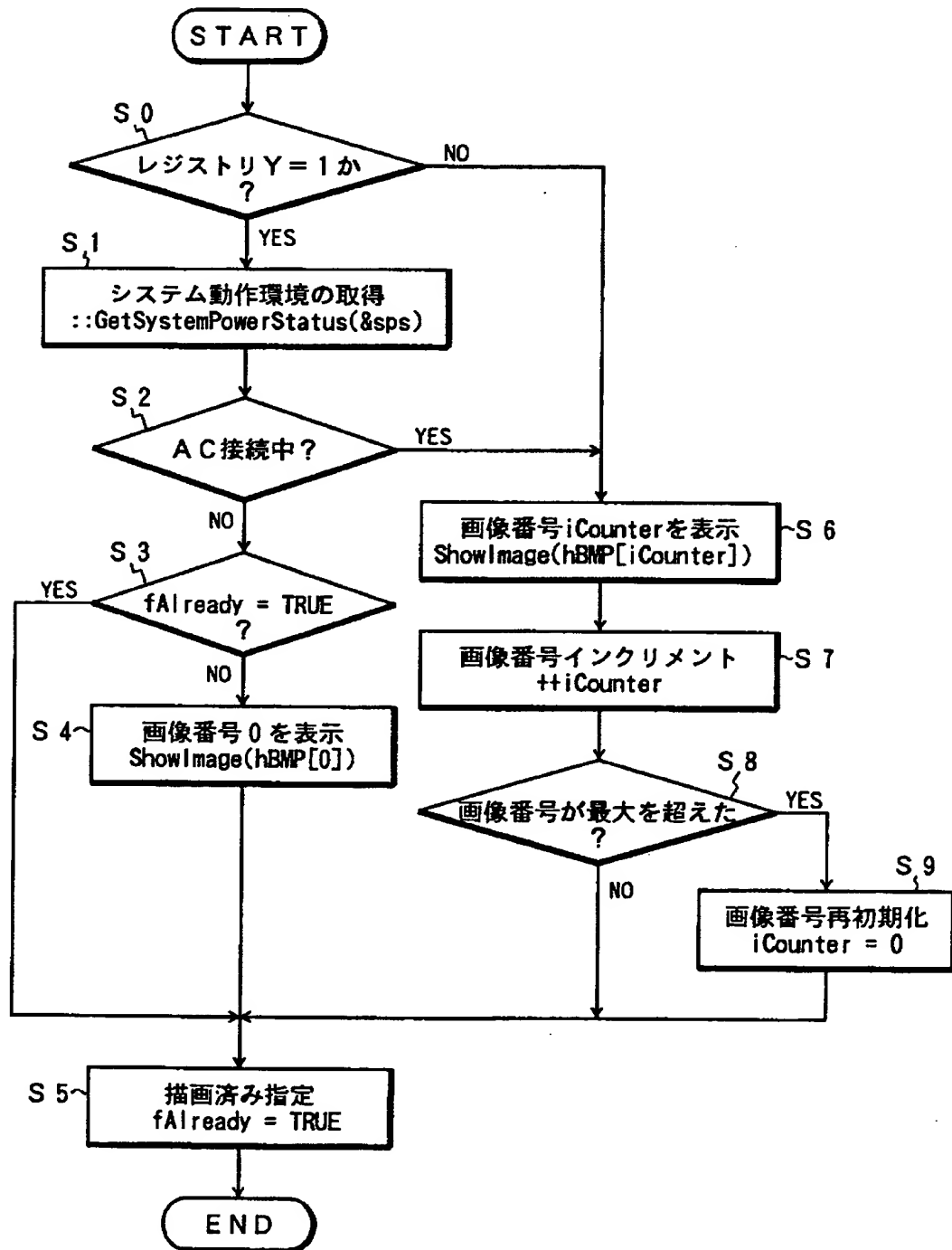
【図 14】

図 11 に従ったCPU負荷切替えの有効と無効を設定するプロパティ画面の
説明図



【図 15】

図 12 のプロパティ画面による設定したレジストリの内容に従った本発明による
アニメーション・ソフトウェア処理のフローチャート



【書類名】 要約書

【要約】

【課題】 ソフトウェアが省電力が必要な環境か否か判断して環境に最適な動作を実現する。

【解決手段】 ソフトウェア処理装置は、動作環境判断部 3 6 でアニメーション等の処理を実行する際に省電力を必要とする動作環境か否かを判断し、切替処理部 4 8 により省電力が必要な環境ではソフトウェアの一部の機能を有効とした簡略化された低負荷な処理を実行し、省電力が不要な環境ではソフトウェアの全ての機能を有効とした高負荷な処理を実行する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名 富士通株式会社

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.